Lecture 4: The GP example

Verena Zuber

15/2/2019

```
library(lme4)
library(lattice)
library(ggplot2)
library(nlme)
```

```
data.chol = read.csv("Cholesterol.txt", sep=" ")
dim(data.chol)
```

```
## [1] 441    6
```

```
head(data.chol)
```

```
##   chol doctor age   bmi agedoc sex
## 1 7.13      1  54 27.39     55   0
## 2 7.70      1  55 29.10     55   0
## 3 7.30      1  56 27.90     55   0
## 4 6.89      1  71 26.67     55   1
## 5 6.90      1  72 26.70     55   1
## 6 7.90      1  73 29.70     55   1
```

1. First part: Fitting the model

Model 1 (individual level)

```
model1 = lm(chol ~ age, data=data.chol)
summary(model1)
```

```
##
## Call:
## lm(formula = chol ~ age, data = data.chol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8971 -0.6206 -0.1105  0.5693  2.9456
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.798691   0.268571   10.42   <2e-16 ***
## age         0.051262   0.004301   11.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8362 on 439 degrees of freedom
## Multiple R-squared:  0.2445, Adjusted R-squared:  0.2428
## F-statistic: 142.1 on 1 and 439 DF,  p-value: < 2.2e-16
```

```
data.chol$PooledPredictions = fitted(model1)
plot(data.chol$age,data.chol$chol, xlab = "age", ylab = "total cholesterol", cex = 1.2)
abline(model1, col = "red", lwd =2)
```
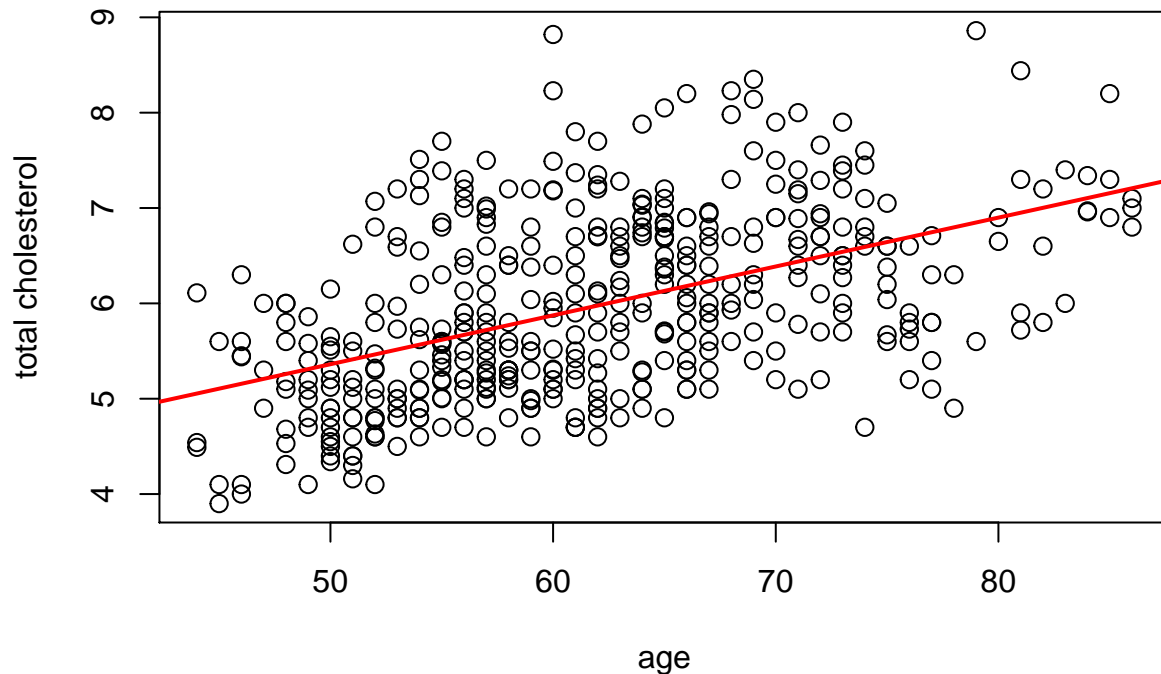
Model 2 (group level)

```r
chol.group = tapply(data.chol$chol,INDEX=data.chol$doctor,FUN=mean)
age.group = tapply(data.chol$age,INDEX=data.chol$doctor,FUN=mean)
Group.Model = lm(chol.group ~ age.group)
summary(Group.Model)
```

```
##
## Call:
## lm(formula = chol.group ~ age.group)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7216 -0.4513 -0.1844  0.3020  1.3576
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.30687    5.05233   0.259    0.801
## age.group    0.07548    0.08176   0.923    0.378
##
## Residual standard error: 0.67 on 10 degrees of freedom
## Multiple R-squared:  0.07854,    Adjusted R-squared:  -0.0136
## F-statistic: 0.8524 on 1 and 10 DF,  p-value: 0.3776
```

```r
plot(age.group, chol.group,  xlab = "age", ylab = "total cholesterol", cex = 1.2)
abline(Group.Model, col = "red", lwd =2)
```
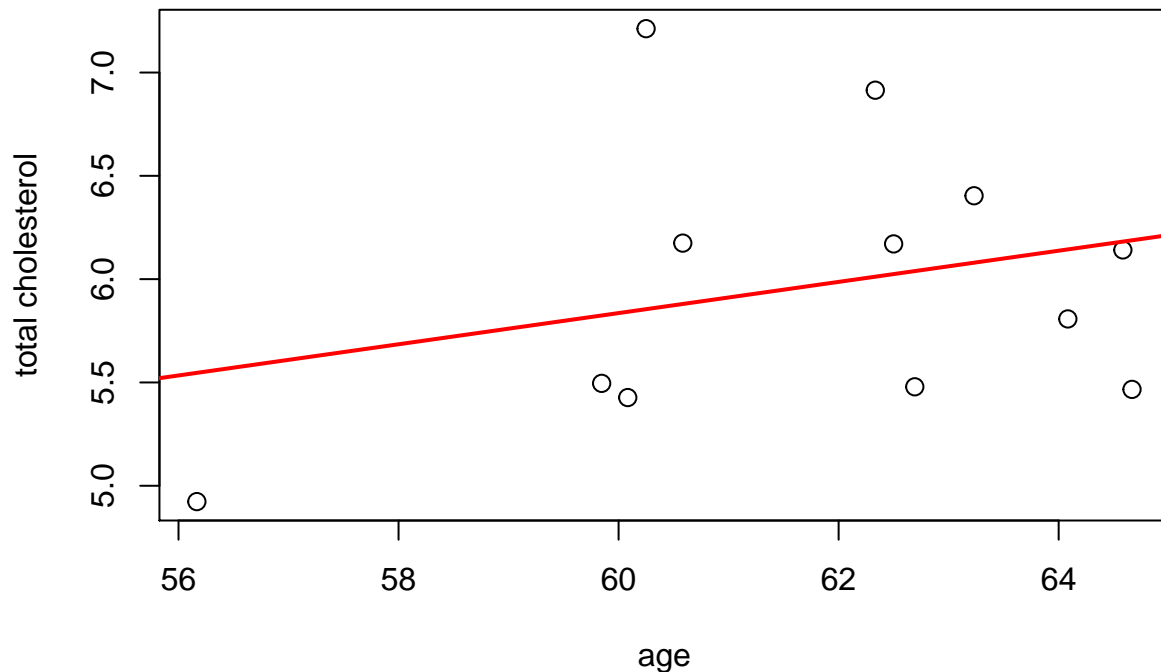
Model 3 (individual regression, one regression model for each group)

```
fits = lmList(chol ~ age | doctor, data=data.chol)
fits
```

```
## Call:
##   Model: chol ~ age | doctor
##    Data: data.chol
##
## Coefficients:
##     (Intercept)         age
## 1     4.3314316 0.04143871
## 2     5.4010239 0.03006138
## 3     3.5442585 0.04202075
## 4     0.7656213 0.08916495
## 5     3.0816670 0.04737634
## 6     2.9456065 0.05329581
## 7     0.1915603 0.08862860
## 8     4.6621327 0.01243695
## 9     2.7119802 0.04830460
## 10    1.5642149 0.06244402
## 11    2.5339182 0.04815023
## 12    2.7996834 0.03780979
##
## Degrees of freedom: 441 total; 417 residual
## Residual standard error: 0.5597412
```

```
xyplot(chol ~ age| doctor, data=data.chol, type = c("p", "r"),
 col.line = "darkorange", xlab = "age", ylab = "total cholesterol")
```
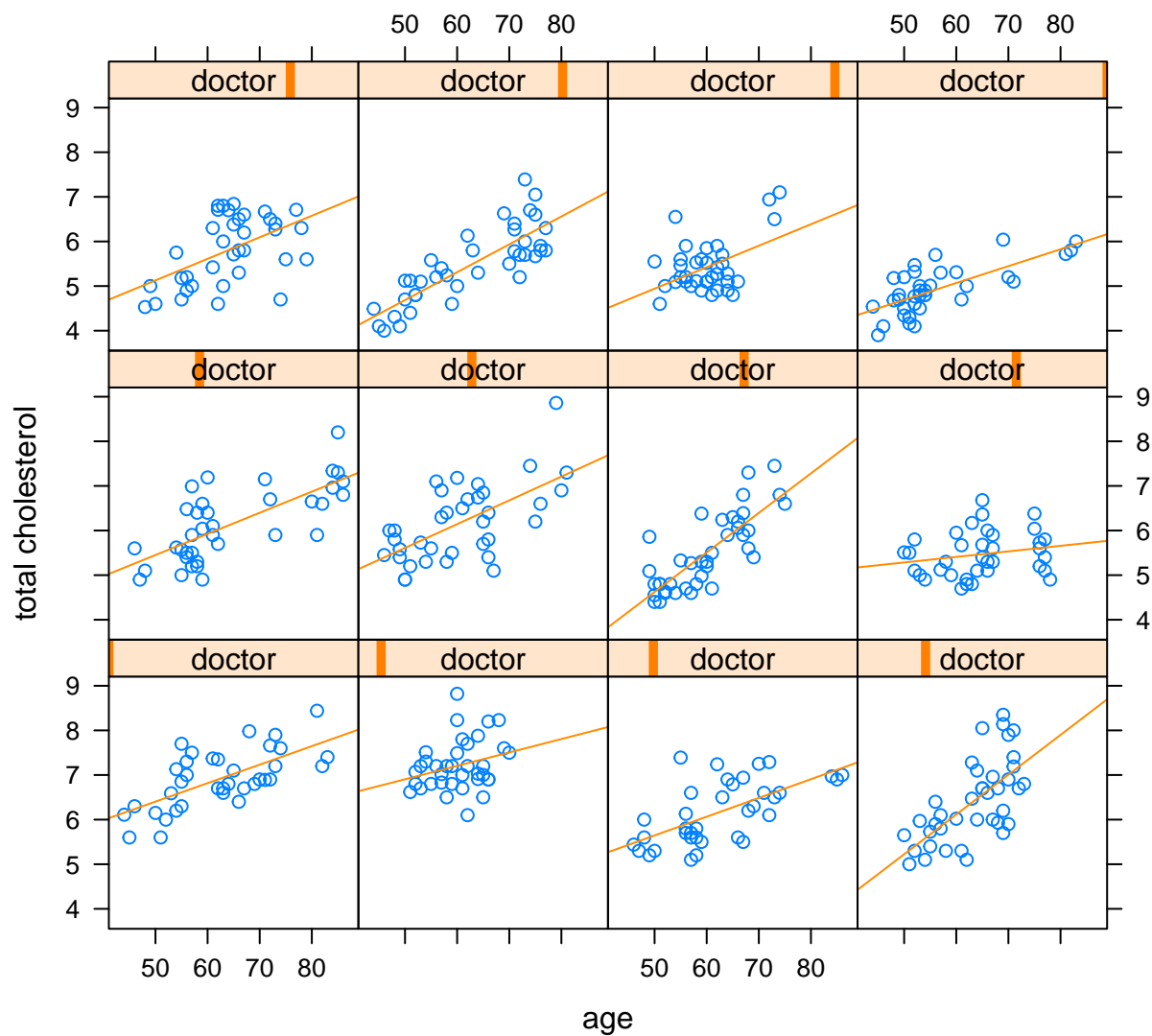
Model 4 (fixed effects)

```
FE.Model = lm(chol ~ age + as.factor(doctor), data=data.chol)
summary(FE.Model)
```

```
##
## Call:
## lm(formula = chol ~ age + as.factor(doctor), data = data.chol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59881 -0.40321 -0.08463  0.37929  1.77313
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.826236   0.213854  17.892  < 2e-16 ***
## age                 0.049543   0.003065  16.164  < 2e-16 ***
## as.factor(doctor)2  0.400993   0.136014   2.948  0.00337 **
## as.factor(doctor)3 -0.752146   0.135865  -5.536 5.41e-08 ***
## as.factor(doctor)4 -0.555317   0.133254  -4.167 3.73e-05 ***
```

```
## as.factor(doctor)5  -0.884528    0.136039  -6.502 2.21e-10 ***
## as.factor(doctor)6  -0.653299    0.135970  -4.805 2.15e-06 ***
## as.factor(doctor)7  -1.295580    0.133444  -9.709  < 2e-16 ***
## as.factor(doctor)8  -1.563657    0.136053 -11.493  < 2e-16 ***
## as.factor(doctor)9  -1.193645    0.135970  -8.779  < 2e-16 ***
## as.factor(doctor)10 -1.453255    0.133231 -10.908  < 2e-16 ***
## as.factor(doctor)11 -1.376027    0.136039 -10.115  < 2e-16 ***
## as.factor(doctor)12 -1.685593    0.137173 -12.288  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5764 on 428 degrees of freedom
## Multiple R-squared:   0.65,  Adjusted R-squared:  0.6402
## F-statistic: 66.24 on 12 and 428 DF,  p-value: < 2.2e-16
```

Model 5 (random intercept)

```
RandomIntercept = lme( chol ~ age, random = ~ 1 | doctor, data = data.chol )
summary(RandomIntercept)
```

```
## Linear mixed-effects model fit by REML
##  Data: data.chol
##       AIC      BIC    logLik
##   828.697 845.035 -410.3485
##
## Random effects:
##  Formula: ~1 | doctor
##         (Intercept)  Residual
## StdDev:   0.6347908 0.5764246
##
## Fixed effects: chol ~ age
##                 Value  Std.Error  DF  t-value p-value
## (Intercept) 2.9060357 0.26477408 428 10.97553       0
## age         0.0495831 0.00306279 428 16.18888       0
##  Correlation:
##     (Intr)
## age -0.714
##
## Standardized Within-Group Residuals:
##        Min          Q1        Med         Q3        Max
## -2.7850636 -0.7012544 -0.1419474  0.6536599  3.0850909
##
## Number of Observations: 441
## Number of Groups: 12
```

Model 6 (random intercept and slope)

```
RandomSlope = lme( chol ~ age, random = ~ 1+age | doctor, data = data.chol )
summary(RandomSlope)
```

```
## Linear mixed-effects model fit by REML
##  Data: data.chol
##       AIC      BIC    logLik
##   821.9886 846.4956 -404.9943
##
## Random effects:
```

```
##  Formula: ~1 + age | doctor
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev     Corr
## (Intercept) 1.28163791 (Intr)
## age         0.01771585 -0.872
## Residual    0.55997509
##
## Fixed effects: chol ~ age
##                 Value Std.Error  DF  t-value p-value
## (Intercept) 2.8791744 0.4215200 428 6.830458       0
## age         0.0500704 0.0060597 428 8.262837       0
##  Correlation:
##     (Intr)
## age -0.901
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -2.8523390 -0.6664557 -0.1141926  0.6206844  3.0809630
##
## Number of Observations: 441
## Number of Groups: 12
```

2. Second part: Using ggplot

In order to use ggplot we need to add the model fits (fitted()) from the random effects (Model 5-7) to the data sets.

```
data.chol$FixedEffects = fitted(RandomIntercept)
data.chol$VaryingInterceptPredictions = fitted(RandomIntercept)
data.chol$VaryingSlopePredictions = fitted(RandomSlope)
```

Pooled prediction and fixed effects

```
gg.fe <- ggplot(data.chol, aes(x = age, y = chol, group = doctor)) +
    geom_line(aes(y = PooledPredictions), color = "darkgrey") +
    geom_line(aes(y = FixedEffects), color = "green") +
    geom_point(alpha = 0.3, size = 2) +
    xlab("age") + ylab("total cholesterol") +
    facet_wrap(~doctor) +
    theme_bw()

print(gg.fe)
```
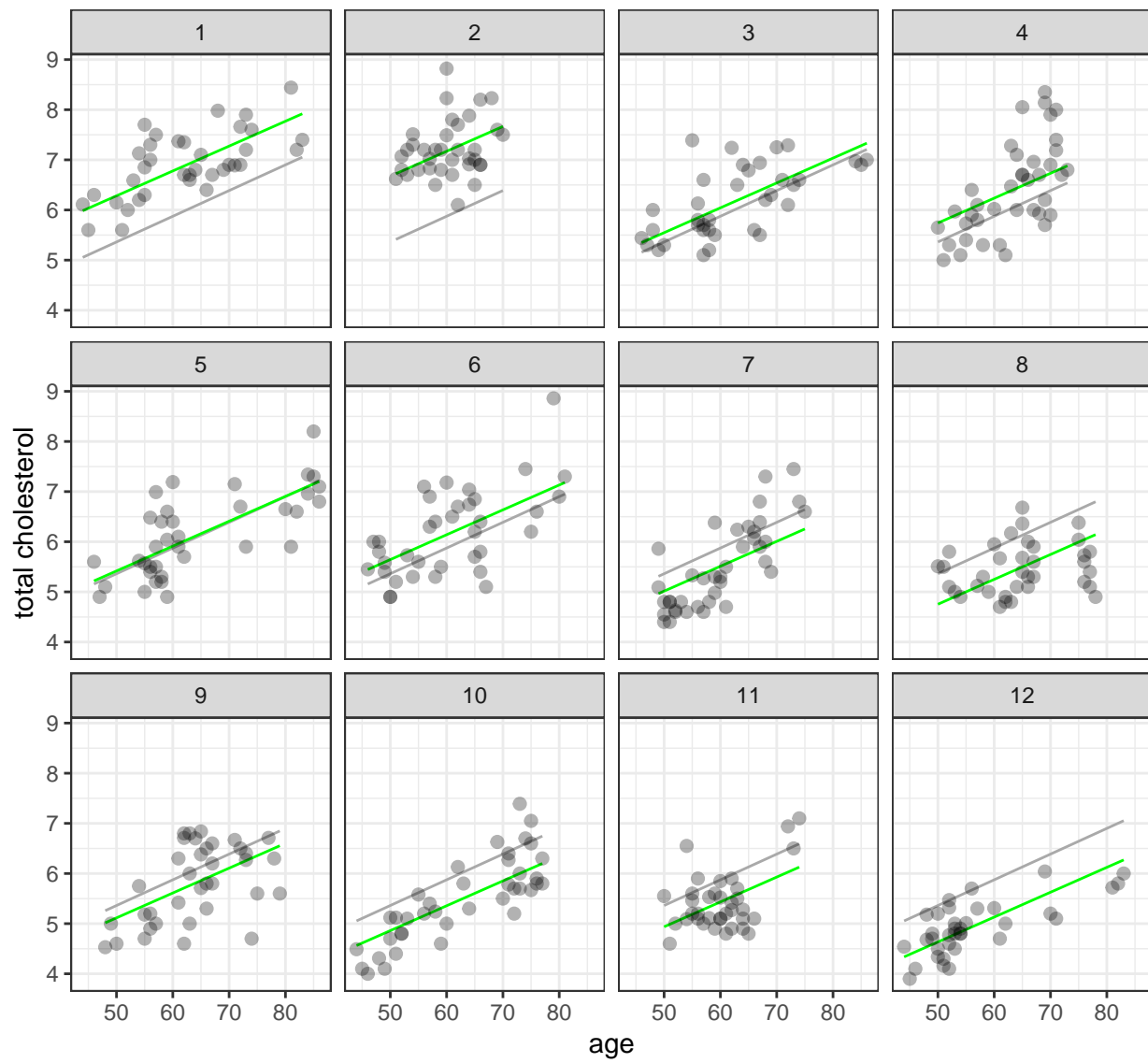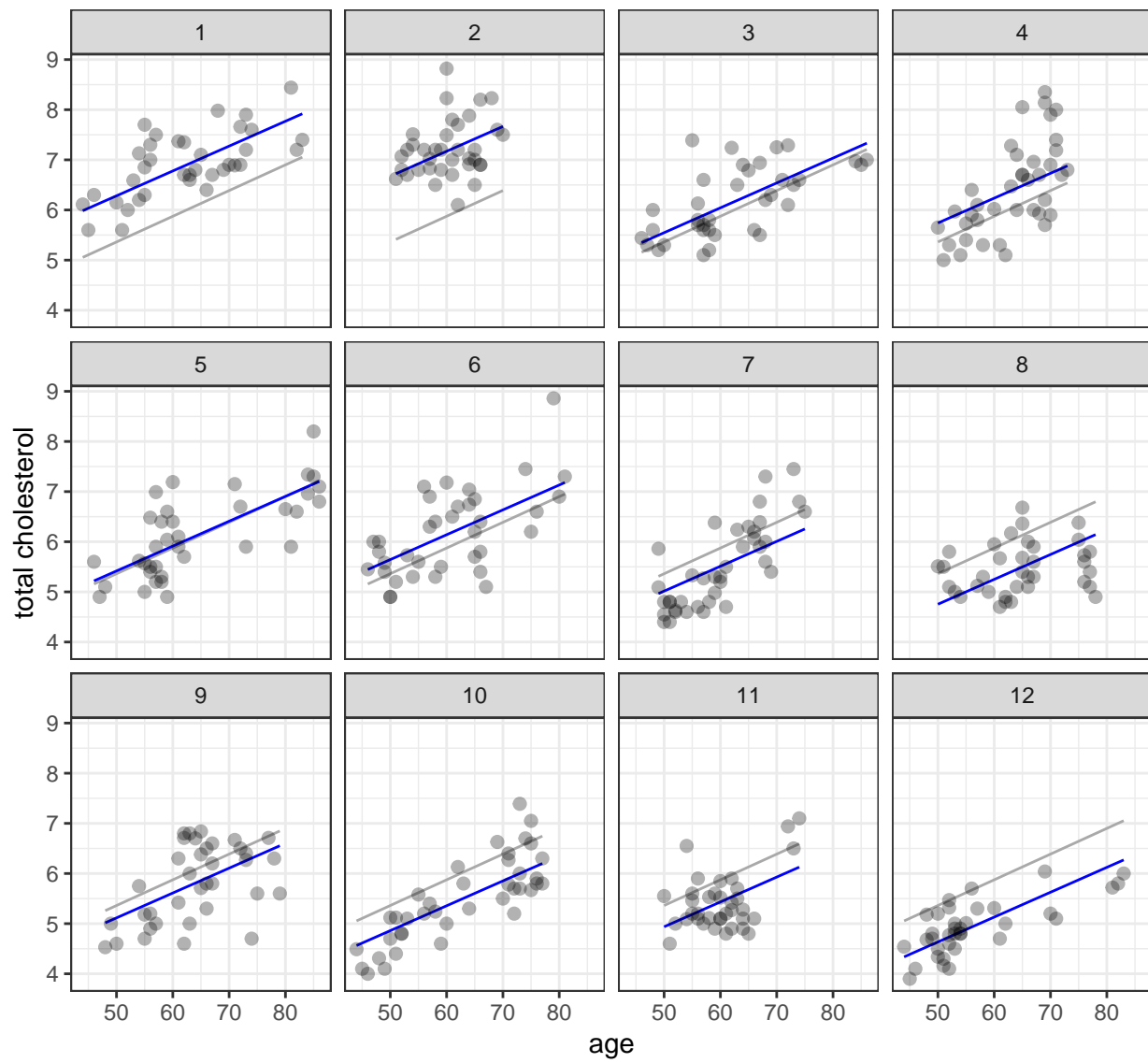
Pooled prediction and random intercept

```
gg.intercept <- ggplot(data.chol, aes(x = age, y = chol, group = doctor)) +
    geom_line(aes(y = PooledPredictions), color = "darkgrey") +
    geom_line(aes(y = VaryingInterceptPredictions), color = "blue") +
    geom_point(alpha = 0.3, size = 2) +
    xlab("age") + ylab("total cholesterol") +
    facet_wrap(~doctor) +
    theme_bw()

print(gg.intercept)
```

Pooled prediction and random intercept and slope

```r
gg.slope <- ggplot(data.chol, aes(x = age, y = chol, group = doctor)) +
    geom_line(aes(y = PooledPredictions), color = "darkgrey") +
    geom_line(aes(y = VaryingSlopePredictions), color = "red") +
    geom_point(alpha = 0.3, size = 2) +
    xlab("age") + ylab("total cholesterol") +
    facet_wrap(~doctor) +
    theme_bw()

print(gg.slope)
```